

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization International Bureau



(43) International Publication Date
31 December 2003 (31.12.2003)

PCT

(10) International Publication Number
WO 2004/002109 A1

(51) International Patent Classification⁷: H04L 29/06, 12/56

(21) International Application Number: PCT/IB2003/002409

(22) International Filing Date: 20 June 2003 (20.06.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data: 02014116.4 24 June 2002 (24.06.2002) EP

(71) Applicant (for all designated States except US): INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): BOWEN, Hoyt, Edwin, Jr. [US/US]; 1203 Balmoral Drive, Cary, NC 27511

(US). DROZ, Patrick [CII/CII]; Alte Landstrasse 105, CH-8803 Rueschlikon (CH). KENCL, Lukas [CZ/GB]; 122 Richmond Road, Cambridge, CB4 3PT (GB). METZLER, Bernhard [DE/CH]; Stapferstrasse 21, CH-8006 Zurich (CH).

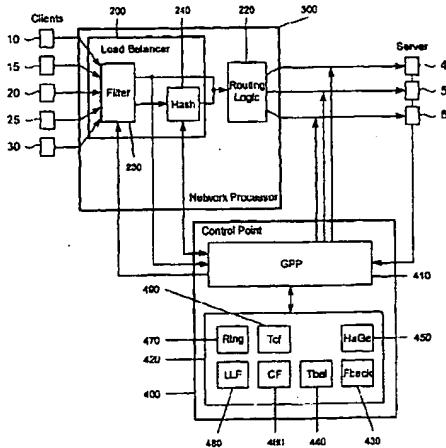
(74) Agent: KLETT, Peter, Michael; International Business Machines Corporation, Saeumerstrasse 4 / Postfach, CH-8803 Rueschlikon (CH).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,

[Continued on next page]

(54) Title: LOAD BALANCING IN DATA NETWORKS



(57) Abstract: Load balancing apparatus for a data communications network comprises hash logic for computing a hash function on incoming data packets. A threshold detector is connected to the hash logic for triggering, in response to utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic of parameters of the hash function from a first set of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects. In use, the hash logic, directs the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the first set of parameters, and, if the threshold is exceeded, selectively directs the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via the selected one of the first and second routing paths based on the results of one of the separate hash computations.

WO 2004/002109 A1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization International Bureau



(43) International Publication Date
31 December 2003 (31.12.2003)

PCT

(10) International Publication Number
WO 2004/002109 A1

(51) International Patent Classification⁷: H04L 29/06, 12/56

(74) Agent: KLETT, Peter, Michael; International Business Machines Corporation, Saeumerstrasse 4 / Postfach, CH-8803 Rueschlikon (CH).

(21) International Application Number:

PCT/IB2003/002409

(22) International Filing Date: 20 June 2003 (20.06.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
02014116.4 24 June 2002 (24.06.2002) EP

(71) Applicant (for all designated States except US): INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NJ 10504 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): BOWEN, Hoyt, Edwin, Jr. [US/US]; 1203 Balmoral Drive, Cary, NC 27511 (US). DROZ, Patrick [CH/CH]; Alte Landstrasse 105, CH-8803 Rueschlikon (CH). KENCL, Lukas [CZ/GB]; 122 Richmond Road, Cambridge, CB4 3PT (GB). METZLER, Bernhard [DE/CH]; Stapsstrasse 21, CH-8006 Zurich (CH).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

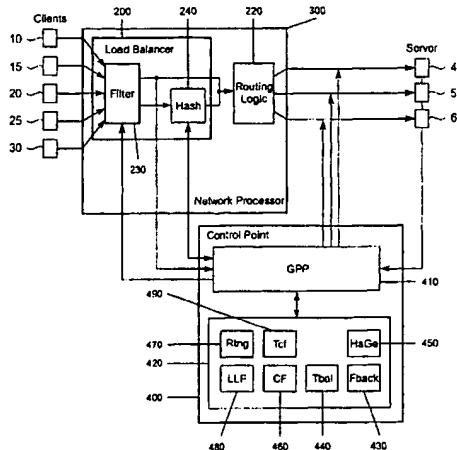
(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: LOAD BALANCING IN DATA NETWORKS



(57) Abstract: Load balancing apparatus for a data communications network comprises hash logic for computing a hash function on incoming data packets. A threshold detector is connected to the hash logic for triggering, in response to utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic of parameters of the hash function from a first set of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects. In use, the hash logic, directs the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the first set of parameters, and, if the threshold is exceeded, selectively directs the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via the selected one of the first and second routing paths based on the results of one of the separate hash computations.

WO 2004/002109 A1



SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(15) Information about Correction:

see PCT Gazette No. 12/2004 of 18 March 2004, Section II

Published:

— *with international search report*

(48) Date of publication of this corrected version:

18 March 2004

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

LOAD BALANCING IN DATA NETWORKS

The present invention generally relates to load balancing in data networks and particularly relates to a method and apparatus for load balancing in a data network.

5 A data network typically comprises a plurality of server computers interconnected to a plurality of client computers via a data communications network infrastructure. The network infrastructure typically comprises a plurality of intermediate data communications devices or nodes such as switches, routers
10 and the like for routing data packets between the client computer and the server computers. Such data communications devices typically comprise a plurality of input/output (I/O) ports, a switch fabric for routing data packets received on one port to one or more of the other ports, and control logic
15 for controlling the switch fabric to make appropriate connections between the ports based on address information contained in the transient data packets.

A problem associated with such data networks is that of balancing loads between different nodes within the network
20 infrastructure. As the amount of data traffic handled by the network infrastructure increases, the balance between loads carried by the nodes in the infrastructure becomes harder to maintain in a timely manner. Accordingly, communication bottlenecks and delays are incurred.

25 In accordance with the present invention, there is now provided .Load balancing apparatus for a data communications network, the apparatus comprising: hash logic for computing a hash function on incoming data packets; a threshold detector connected to the hash logic for triggering, in response to
30 utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic of parameters of the hash function from a first set of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects; wherein, the hash logic, in use, directs

the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the first set of parameters, and, if the threshold is exceeded, for selectively directing the packets to one of the first 5 routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via the selected one of the first and second routing paths based on the results of one of the separate hash computations.

- 10 Preferably, the hash logic in use directs the data packet to the first routing path if the results of the separate hash computations intersect and otherwise directs the data packet to the second routing path. In a preferred embodiment of the present invention, the apparatus further comprises a filter
- 15 connected to the hash logic for selectively bypassing the hash logic for flows having a lifetime exceeding a predefined value. In a particularly preferred embodiment of the present invention, the apparatus further comprises the first routing path and the second routing path, the first routing path
- 20 comprising first routing logic connected to the hash logic, and the second routing path comprising second routing logic connected to the hash logic, wherein the first routing path is faster than the second routing path, and wherein, on the second routing path, downstream objects are selected based on
- 25 packet flow status.

The first routing logic may comprise at least one network processor and the second routing logic may comprise at least one general purpose processor. The second routing logic may be configured to detect a flow delimiter in a flow of data

- 30 packets and, on detection of the start indicator, to route the corresponding flow according to the hash computation using the second parameters. The second routing logic may also be configured to detect flows of packets exceeding a predetermined inactivity time and to route such flows
- 35 according to the hash computation using the second parameters. Further, the second routing logic may be configured to detect

flows of packets exceeding a predetermined lifetime and to direct such flows to the first routing logic.

The present invention extends to an application specific
5 integrated circuit comprising load balancing apparatus as herein before described. The present invention also extends to a network infrastructure node comprising load balancing apparatus as herein before described. Furthermore, the present invention extends to a data communications network comprising
10 such a network infrastructure node.

Viewing the present invention from another aspect, there is now provided a method of load balancing in a data communications network, the method comprising: computing a hash function on incoming data packets; triggering, in
15 response to utilization of the downstream objects exceeding a predefined threshold, redefinition of parameters of the hash function from a first set of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects; and, directing the packets for routing to
20 downstream objects in the network via a first routing path based on a hash computation using the first set of parameters, and, if the threshold is exceeded, selectively directing the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the
25 first and the second sets of parameters for subsequent routing of the packets via the selected one of the first and second routing paths based on the results of one of the separate hash computations.

In a preferred embodiment of the present invention, the method
30 comprises directing the data packets to the first routing path if the separate hash computations intersect and otherwise directing the data packets to the second routing path.

To best balance 10,000s and 100,000s of flows in hardware, it is necessary to determine a scheme that keeps minimal state on

flows, but maintains the connectivity of active flows such as TCP flows between hosts.

In a preferred embodiment of the present invention there is provided an iterative hardware load balancing technique in which a hash function is employed to balance packet flows between a number of hosts, intermediate nodes, and/or network interfaces in a data communications network. The hash is performed on a portion of each packet that is constant for the duration of a flow, such as the address of the source 5 associated with the flow. The technique alternates between a state in which the hash function is computed with only one set of hash parameters and a state in which two hash parameter sets are given. In one state, one hash result is known. In the other state, the difference between two hash results is known. 10 15 The technique identifies a routing fast path and a routing slow path. The routing fast path may be performed in dedicated hardware such as Network Processors or similar application specific integrated circuits (ASICs). The routing slow path may be better performed in software on a general purpose 20 processor (GPP). No data flows in progress are moved between load balanced objects unless continuity is maintained. This advantageously insures that flow connectivity is uninterrupted and that packets are not reordered. In addition, state is retained for data flows outside the intersection of the two 25 hash functions. This kept retained state is however constantly and aggressively reduced. This advantageously minimizes hardware costs. Furthermore, by performing the fast path routing via dedicated hardware such as network processor hardware, data communications performance is improved beyond 30 hitherto available levels.

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a data communications network;

Figure 2 is a block diagram of an infrastructure node of the data communications network;

Figure 3 is another block diagram of the infrastructure node;

5 Figure 4 is a flow chart associated with a control point of the infrastructure node;

Figure 5 is a flow chart associated with a load balancer of the infrastructure node; and,

10 Figure 6 is yet another flow chart associated with a load balancer of the infrastructure node.

Referring first to Figure 1, a data communications network 140 comprises a plurality of network objects 10-120. The network objects include network nodes 10-90 and intervening network 15 links 100-120 for communicating data packets between the network nodes 10-30. The network nodes 10-90 include end point nodes 10-60 and infrastructure nodes 70-90. The infrastructure nodes 70-90 of the links 110-130 collectively form a network infrastructure 130. The end point nodes 10-60 include a 20 plurality of client data processing devices 10-30 and a plurality of server computer systems 40-60. The clients 10-30 and the servers 40-60 are interconnected via an intervening network infrastructure 100. Each client 10-30 may be a personal computer, laptop computer, personal digital 25 assistant, mobile telephone, or the like. Each server 40-60 may be a file server, print server, or similar host data processing or data storage resource.

In operation, data communications between the clients 10-30 and the servers 40-60 is effected via flows of data packets 30 through the network infrastructure 100. The flows of data packets are passed between the infrastructure nodes 70-90 of the network infrastructure en route between the clients 10-30

and the servers 40-60 according to one or more data communications protocol. Each infrastructure node 70-90 performs a packet routing function to forward data packets it receives to an appropriate recipient object

- 5 Referring now to Figure 2, each infrastructure node 70-90 comprises first routing logic 220, second routing logic 210, and a load balancer 200 connected to both the first routing logic 220 and the second routing logic 210. Both the first routing logic 220 and the second routing logic 210 perform the same routing function. However, the first routing logic 220 performs the routing function faster than the second routing logic 210. The first routing logic 220 thus provides a relatively fast routing path and the second routing logic 210 provides a relatively slow routing path.
- 10
- 15 Referring now to Figure 3, in a particularly preferred embodiment of the present invention, each infrastructure node comprises a network processor (NP) 300 connected to a control point 400. Both the load balancer 200 and the first routing logic 220 are implemented in the NP 300. Specifically, the NP 20 comprises executable software for performing load balancing decisions of the load balancer 200, together with a hardwired logic implementation of first routing logic 220 for performing the packet routing function. The load balancer 200 comprises a filter 230 and hash logic 240 connected to the filter 230.
- 25

A control point (CP) 400 connected to the NP 300 comprises a general purpose processing unit (GPP) 410 connected to a memory 420. In the memory 420 is stored computer program code for implementing, on execution by the GPP 410, a flow state controller 430, threshold detectors 440 and 490, a hash parameter generator 450, a conflicting flow state table 460, a 'long living' flows state table 480, and a router 470. The flow state controller 430 computes state for client-server flows. The threshold detector 440 monitors the current load status of the servers 40-60 to provide load management. The

30

35

router 470 implements the second routing logic 210. Performance of the packet routing function by the CP 400 is slower than performance of the packet routing function by the NP 300.

5

In operation, the load balancer 200 determines whether each flow of incoming data packets is processed by the first routing logic 220 or by the CP 400 in dependence on current status of the filter 230 and the hash logic 240. The status of 10 both the filter 230 and the hash logic 240 is controlled by the CP 400 based on prevailing demands on one or more of the downstream objects in the network 140. Such load balancing is desirable in the interests of, for example, preventing unnecessary overloading of one or more of the downstream 15 objects by non optimally distributed client requests. By preventing unnecessary overloading, the available service level in the network 140 is optimized. The load balancer 200, via the hash logic 240 therein, performs hash functions to balance flows of data packets between objects in the network. 20 Specifically, the hash logic 240 performs the hash functions on some portion of the incoming data packets that is constant for the duration of a packet flow, such as the source address. The load balancer 200 alternates between a state in which the difference between the results of two hash functions is known, 25 and a state in which the result of only one hash function is known. The filter 230 may bypass the hash logic 240 for some packet flows.

In operation, the filter 230 directs incoming flows of data packets either to the first routing logic 220 or the hash 30 logic 240. The hash logic 240, in dependence on a control input from the CP 400, directs the packets either to the first routing logic 220 or to the CP 400 in which the second routing logic 210 is implemented. If the packet is passed to the first routing logic 220, then it is coupled with routing information 35 from either the hash logic 240 or the filter 230. The routing information added directs the first routing logic 220 to choose a specified routing path. If the hash function

redirects a packet to the second routing logic 210, then the routing information is also included in the packet.

Referring now to Figure 4, the control point 400 starts in step 500 with the initialization of both the hash logic 240 and the filter 230 of the load balancer 200. The filter 230 is set to void. The initial parameters of the hash logic 240 may be based on operator configuration. Alternatively, the initial hash parameters may be based on an automated query of resource capacities of objects 40 - 60 to be balanced. Examples of such resource capabilities include CPU speeds, interface speeds, and the like. In step 510, the CP 400 accumulates statistics in the feedback memory 430 relating to resource utilization in the objects 40 - 60 to be balanced.

At step 520, the threshold detector 440 checks the statistics accumulated in the flow state controller 430 against a predetermined threshold to determine if at least one of the objects 40 - 60 is over utilized relative to the other objects. In the event that the determination is positive, then, at step 530, new hash parameters are computed by the hash parameter generator 450 and loaded as a second parameter set into the hash logic 240. The new hash parameters optimally distribute the network traffic based on the statistics gathered. In step 540, the CP 400 acts as the second routing logic 210. Specifically, the CP 400 receives packets from the load balancer 200 which produced different hash results when the hash logic 240 computed the hash function with both the old and the new parameter set. For such 'conflicting flows', the CP 400 establishes and maintains per flow state information within the conflicting flow state table 460. If the CP 400 does not receive packets for a conflicting flow within a configurable time interval, or if the CP 400 at least once receives a flow end indicator such as a TCP FIN bit, then the flow in question is regarded as closed and its state is marked as 'old terminated'.

In step 545, the CP 400 forwards the packet through the second routing logic 470 to one of the load balanced objects 40 - 60. If the status of the flow to which the packet belongs is marked as 'old terminated', then the packet is forwarded to 5 the result of the hash computation using the new hash parameters. Otherwise, the packet is forwarded to the result of the hash computation using the old hash parameters.

After a time interval not shorter than the time interval allocated to detection of flow termination in step 540, the 10 threshold detector 490 checks, at step 550, the number of unterminated conflicting flows against a predetermined threshold. If the threshold detector 490 detects a smaller or equal amount of flows than set as the threshold, then at step 560 all local state of unterminated conflicting flows is moved 15 to the 'long living' flows state table 480 and the conflicting flow state table 460 is cleared. In this case, at step 560, the contents of the 'long living' flows state table 480 is transferred to the filter function 230 of the load balancer 200 and the old hash parameters of the hash logic 240 are 20 removed. If, at step 550, the threshold detector 490 detects more unterminated conflicting flows than set as the threshold, then the CP 400 keeps receiving hash conflicting packets from the load balancer 200.

At step 570, the CP 400 receives copies of packets belonging 25 to 'long living' flows. In step 580, the CP 400 checks to determine if a packet carries a flow end indicator such as a TCP FIN bit or if a 'long living' flow was inactive for a predetermined interval. If both determinations are negative, then the CP 400 returns to step 570, continuing to receive 30 copies of packets belonging to 'long living' flows.

If, at step 580, a flow termination is detected, then, at step 585, the corresponding flow entry is removed from both the 'long living' flow table 480 and the filter 230. At step 590, the CP 400 checks if the 'long living' flow table 480 is 35 empty. If so, the status of the CP 400 returns to the

accumulation of feedback information in step 510. If the 'long living' flow table 480 is not empty, then the CP 400 returns to step 570, receiving copies of packets belonging to 'long living' flows.

- 5 Deep packet processing may be employed by the CP 400 to identify flows in progress which are to be considered as terminated. For example, in some circumstances, File Transfer Protocol (FTP) flows in progress may be routed to a new server without harm.
- 10 Referring now to Figure 5, in operation, at step 600, the filter 230 checks if a filter rule is set for each incoming packet. If yes, then, in step 610, the filter 230 performs a match of a packet flow identifier against the local filter rules. If no filter rule exists or no rule matches, then, in 15 step 620 the packet is forwarded to the hash logic 240. If the packet matches a filter rule, then, based on the matching filter rule, in step 630 the packet is coupled with its routing information. In step 640 the packet is copied to the CP 400 and in step 650 the packet is sent to the first routing 20 logic 220.

With reference to Figure 6, the hash logic 240 alternates between two states, where either one set of hashing parameters is known or one old set and one new set of hashing parameters are known. At step 700, the hash logic 240 computes the hash 25 result based on the new set of parameters. In step 710 the hash logic 240 checks to determine if a set of old hash parameters exists. If no old parameter set exists, then in step 720 the packet and the hash result are passed to the first routing logic 220 to send the packet to the appropriate 30 target object 40 - 60. If an old set of hash parameters exists, then in step 730, the hash result is recomputed based on the old parameter set. In step 740, the hash result is checked to determine if both hash computations yield the same 35 result. If so, then the packet is passed in step 720 to the first routing logic 220 to be routed to the appropriate target

object 40 - 60. If not, then in step 750, the packet and both hash results are passed to the CP 400 for slow path routing.

In the embodiments of the present invention herein before described, each infrastructure 90-110 in the network 130 5 comprises load balancing functionality. However, it will be appreciated that networks 130 may be assembled in which only a subset of infrastructure nodes comprise load balancing functionality.

Load balancing apparatus for a data communications network 10 comprises hash logic for computing a hash function on incoming data packets. A threshold detector is connected to the hash logic for triggering, in response to utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic of parameters of the hash 15 function from a first set of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects. In use, the hash logic, in use, directs the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the 20 first set of parameters, and, if the threshold is exceeded, for selectively directing the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via 25 the selected one of the first and second routing paths based on the results of one of the separate hash computations.

By way of summary, what has been described herein by way of example of the present invention is load balancing apparatus 200 for a data communications network comprises hash logic 240 30 for computing a hash function on incoming data packets. A threshold detector 440 is connected to the hash logic 240 for triggering, in response to utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic 240 of parameters of the hash function from a first 35 set of parameters to a second set of parameters for

redistributing the data packets amongst the downstream objects. In use, the hash logic 240 directs the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the first set 5 of parameters, and, if the threshold is exceeded, selectively directs the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via the selected one of 10 the first and second routing paths based on the results of one of the separate hash computations. It will be appreciated however, that the present invention is not limited to two sets of hash parameters. More than two sets of hash parameters may be employed by the hash logic 240 in other embodiments of the 15 present invention.

CLAIMS

1. Load balancing apparatus for a data communications network, the apparatus comprising:

hash logic for computing a hash function on incoming data
5 packets;

a threshold detector connected to the hash logic for triggering, in response to utilization of the downstream objects exceeding a predefined threshold, redefinition in the hash logic of parameters of the hash function from a first set 10 of parameters to a second set of parameters for redistributing the data packets amongst the downstream objects; wherein,

the hash logic, in use, directs the packets for routing to downstream objects in the network via a first routing path based on a hash computation using the first set of parameters, 15 and, if the threshold is exceeded, for selectively directing the packets to one of the first routing path and a second routing path in dependence on separate hash computations using the first and the second sets of parameters for subsequent routing of the packets via the selected one of the 20 first and second routing paths based on the results of one of the separate hash computations.

2. Apparatus as claimed in claim 1, wherein the hash logic in use directs the data packet to the first routing path if the results of the separate hash computations intersect and 25 otherwise directs the data packet to the second routing path.

3. Apparatus as claimed in claim 1 or claim 2, further comprising a filter connected to the hash logic for selectively bypassing the hash logic for flows having a lifetime exceeding a predefined value.

30 4. Apparatus as claimed in claim 4, further comprising the first routing path and the second routing path, the first

REPLACED BY
ART 34 AMDT

routing path comprising first routing logic connected to the hash logic, and the second routing path comprising second routing logic connected to the hash logic, wherein the first routing path is faster than the second routing path, and 5 wherein, on the second routing path, downstream objects are selected based on packet flow status.

5. Apparatus as claimed in claim 4, wherein the first routing logic comprises at least one network processor and the second routing logic comprises at least one general purpose 10 processor.

6. Apparatus as claimed in claim 4 or claim 5, wherein the second routing logic is configured to detect a flow delimiter in a flow of data packets and, on detection of the start indicator, to route the corresponding flow according to the 15 hash function using the second parameters.

7. Apparatus as claimed in claim 6, wherein the second routing logic is configured to detect flows of packets exceeding a predetermined inactivity time and to route such flows according to the hash function using the second 20 parameters.

8. Apparatus as claimed in claim 7, wherein the second routing logic is configured to detect flows of packets exceeding a predetermined lifetime and to direct such flows to the first routing logic.

25 9. An application specific integrated circuit comprising load balancing apparatus as claimed in any preceding claim.

10. A network infrastructure node comprising load balancing apparatus as claimed in any of claims 1 to 7.

11. A data communications network comprising a network 30 infrastructure node as claimed in claim 9.

REPLACED BY
ART 34 AMDT

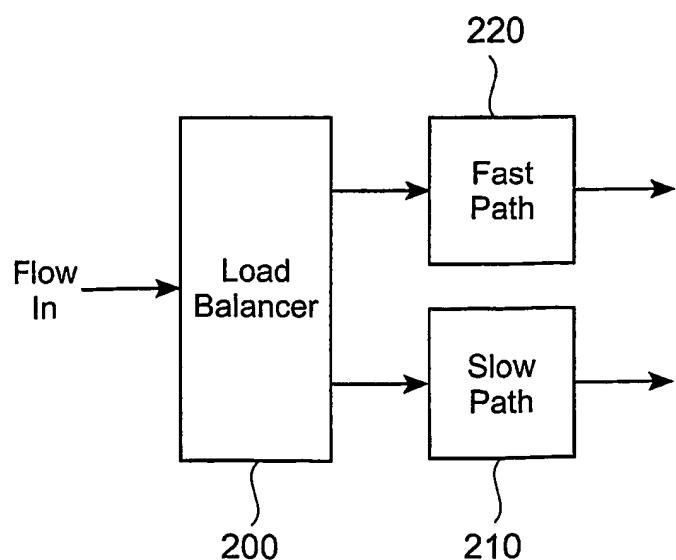
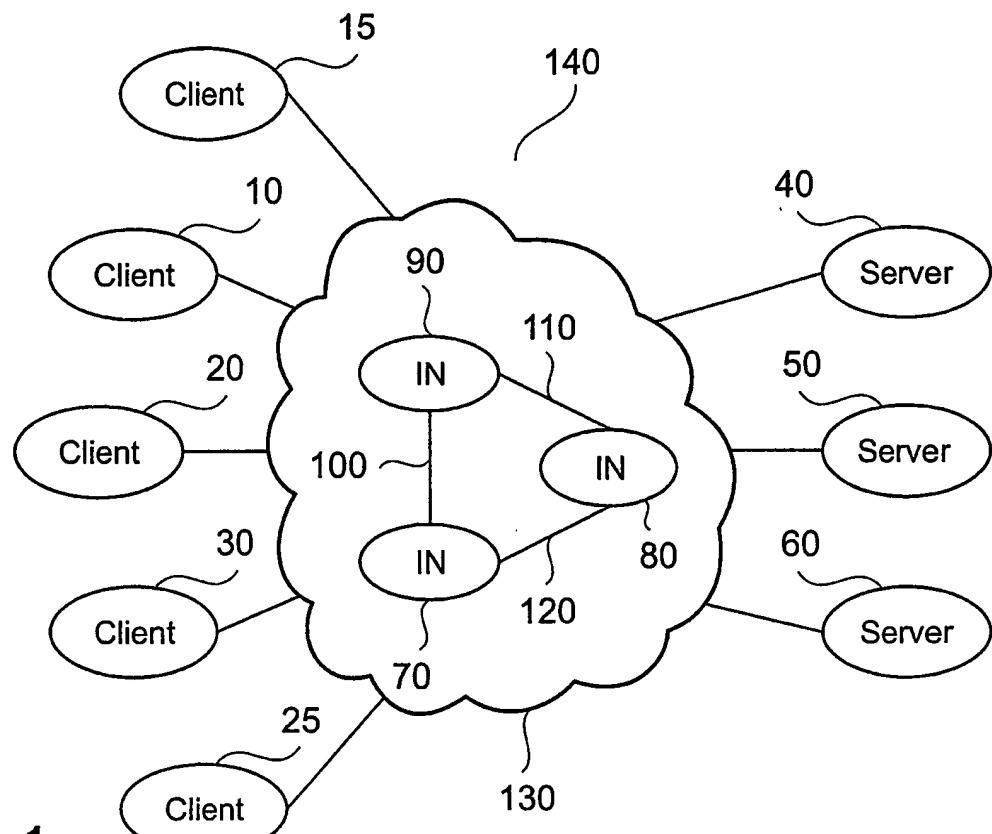
12. A method of load balancing in a data communications network, the method comprising:

computing a hash function on incoming data packets;

triggering, in response to utilization of the downstream
5 objects exceeding a predefined threshold, redefinition of
parameters of the hash function from a first set of parameters
to a second set of parameters for redistributing the data
packets amongst the downstream objects; and,

10 directing the packets for routing to downstream objects in the
network via a first routing path based on a hash computation
using the first set of parameters, and, if the threshold is
exceeded, selectively directing the packets to one of the
first routing path and a second routing path in dependence on
separate hash computations using the first and the second
15 sets of parameters for subsequent routing of the packets via
the selected one of the first and second routing paths based
on the results of one of the separate hash computations.

13. A method as claimed in claim 11, comprising directing the
data packets to the first routing path if the separate hash
20 computations intersect and otherwise directing the data
packets to the second routing path.



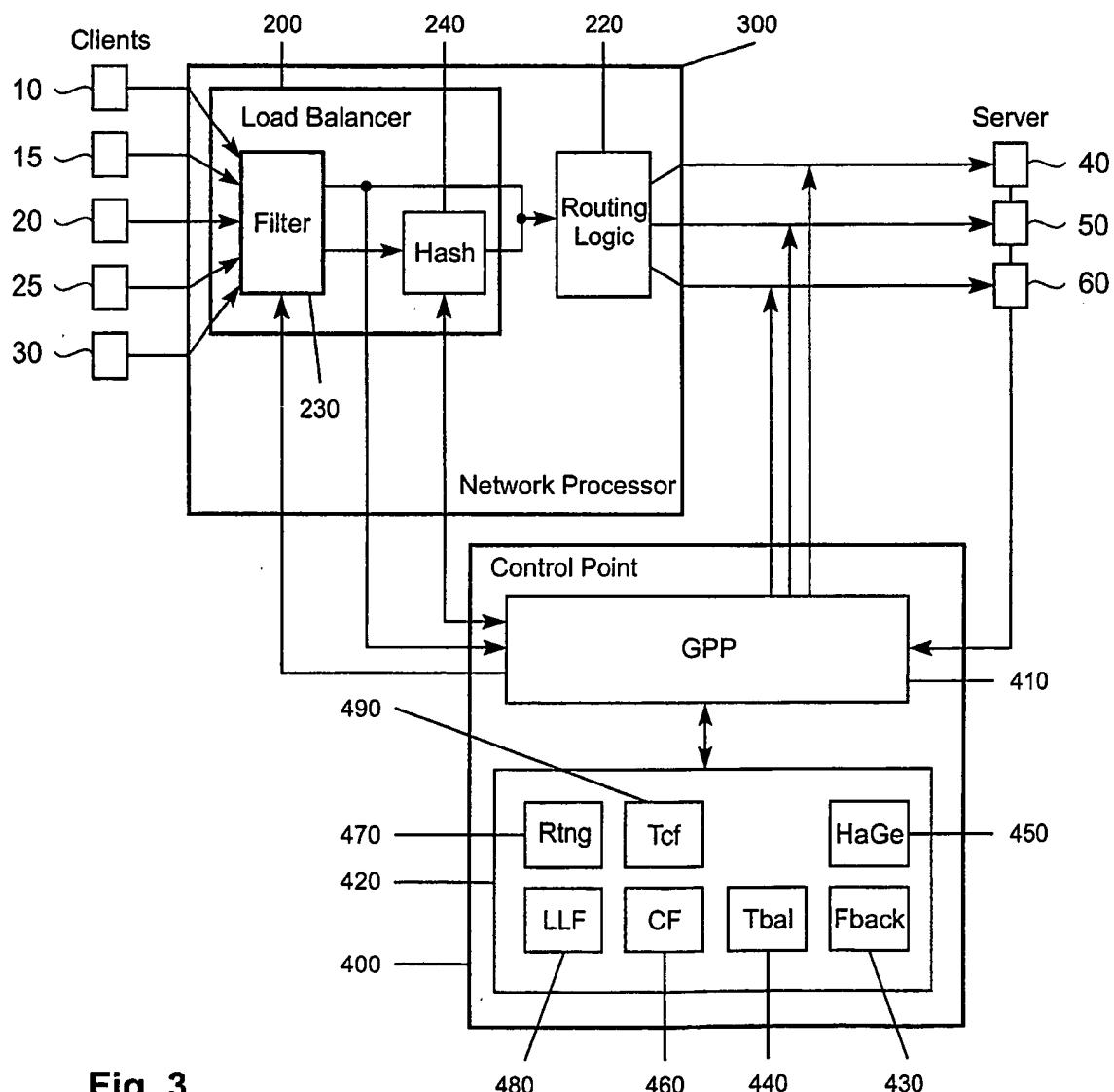


Fig. 3

3/5

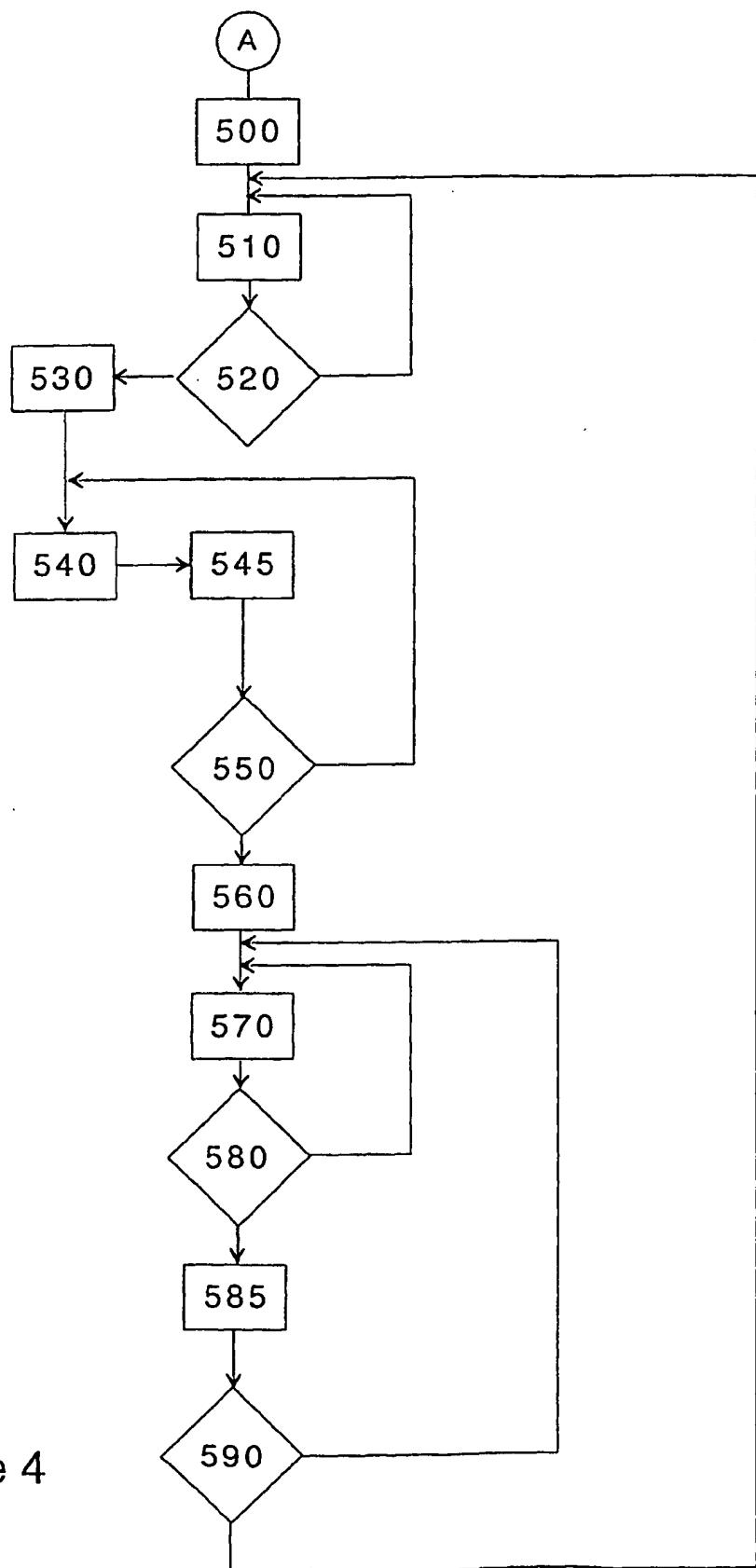


Figure 4

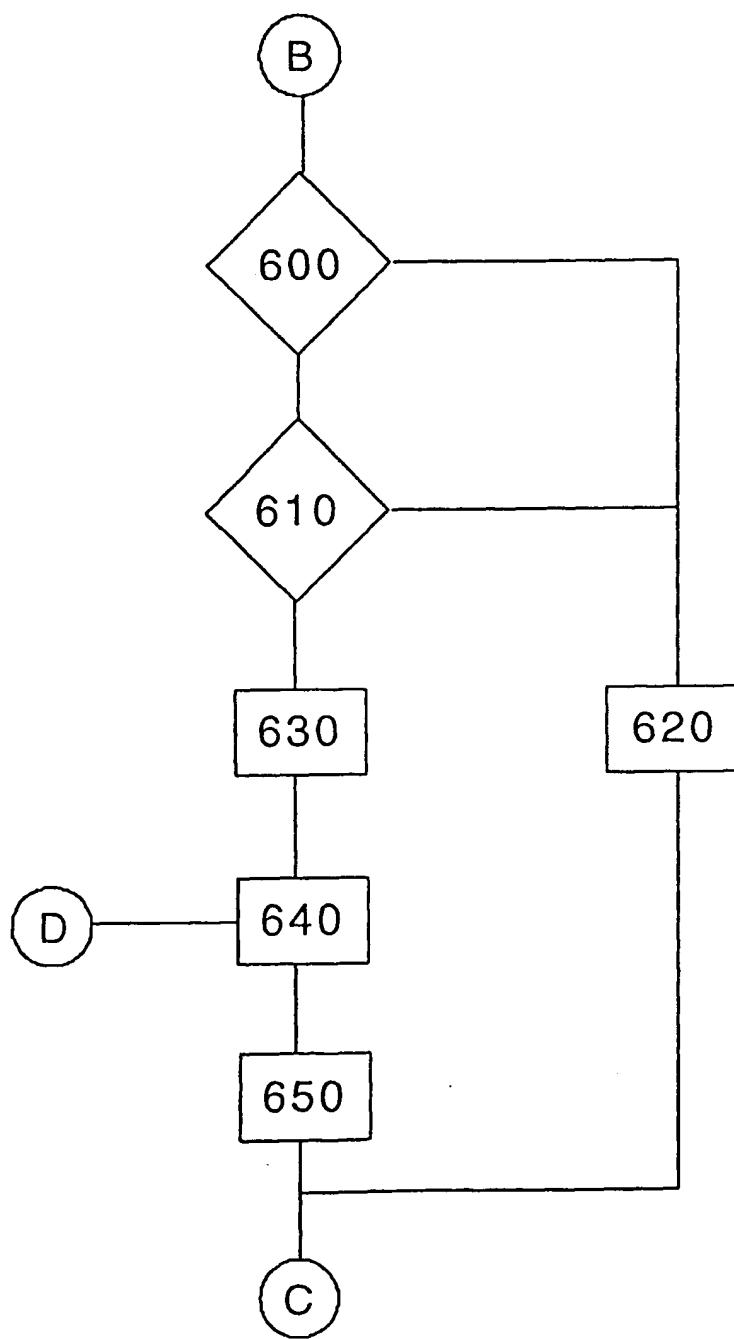


Figure 5

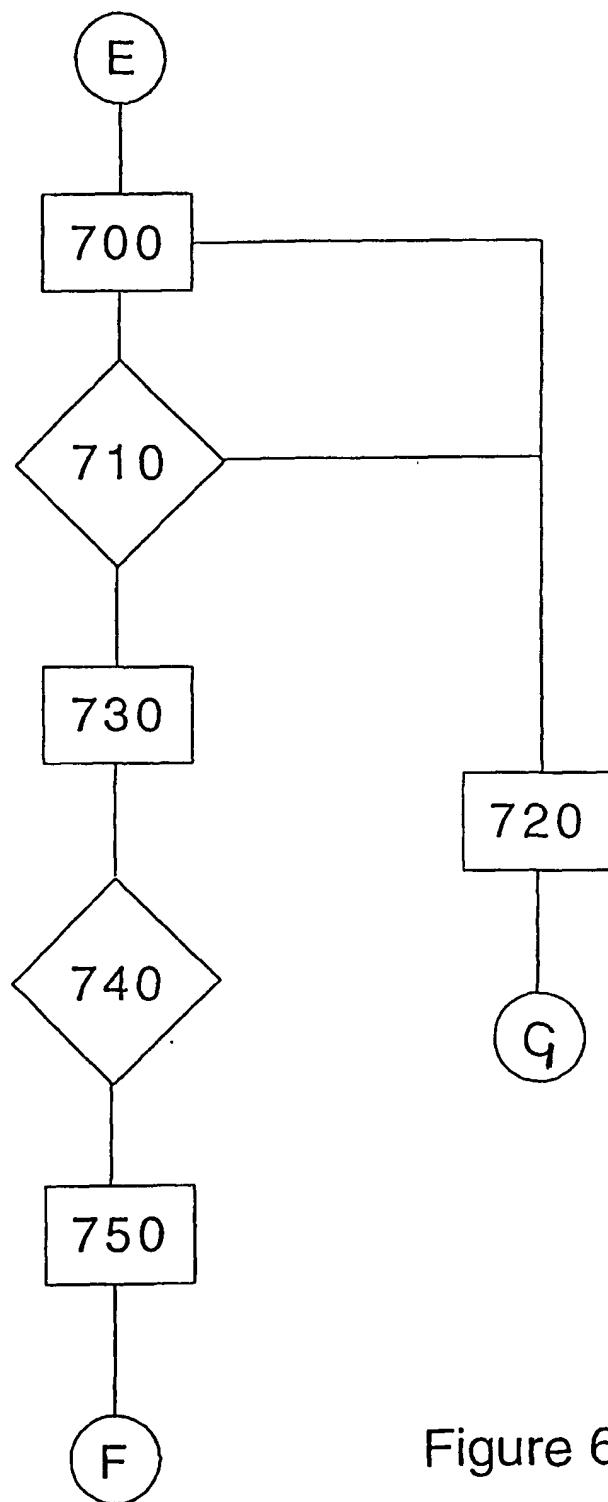


Figure 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB 03/02409A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the International search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category ^a	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 130 849 A (FUJITSU LTD) 5 September 2001 (2001-09-05) page 3, line 2 -page 3, line 10 page 4, line 50 -page 4, line 52 page 15, line 48 -page 16, line 13 claim 36 -----	1-13
X	GB 2 349 296 A (3COM CORP) 25 October 2000 (2000-10-25) page 4, line 14 -page 5, line 18 page 6, line 9 -page 11, line 3 -----	1,2,9-13
Y	US 6 044 080 A (ANTONOV VADIM) 28 March 2000 (2000-03-28) column 2, line 39 -column 3, line 21 column 5, line 45 -column 6, line 34 claims 1,2 -----	1-13

 Further documents are listed in the continuation of box C. Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the International filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the International filing date but later than the priority date claimed

- *T* later document published after the International filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the International search

22 September 2003

Date of mailing of the International search report

06/10/2003

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel: (+31-70) 340-2040, Tx: 81 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Oteo Mayayo, C

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB 03/02409

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 1 128 613 A (LUCENT TECHNOLOGIES INC) 29 August 2001 (2001-08-29) page 2, line 43 -page 3, line 31 page 10, line 26 -page 11, line 31 figures 5,6 -----	1-13
A	WO 02 09307 A (PLURIS INC) 31 January 2002 (2002-01-31) page 3, line 13 -page 5, line 2 page 8, line 5 -page 8, line 23 page 11, line 10 -page 12, line 3 -----	1-13
A	EP 0 903 901 A (FUJITSU LTD) 24 March 1999 (1999-03-24) column 7, line 46 -column 10, line 18 figures 2,3A,3B -----	1-13

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No
PCT/IB 03/02409

Patent document cited in search report		Publication date		Patent family member(s)		Publication date
EP 1130849	A	05-09-2001	JP EP US	2001320420 A 1130849 A2 2001037401 A1	16-11-2001 05-09-2001 01-11-2001	
GB 2349296	A	25-10-2000		NONE		
US 6044080	A	28-03-2000	AU WO	7305698 A 9823127 A1	10-06-1998 28-05-1998	
EP 1128613	A	29-08-2001	CA EP	2328011 A1 1128613 A2	18-07-2001 29-08-2001	
WO 0209307	A	31-01-2002	US AU EP WO US	6553005 B1 6181201 A 1303923 A1 0209307 A1 2002114276 A1	22-04-2003 05-02-2002 23-04-2003 31-01-2002 22-08-2002	
EP 0903901	A	24-03-1999	JP JP EP US	3369445 B2 11096128 A 0903901 A2 6259705 B1	20-01-2003 09-04-1999 24-03-1999 10-07-2001	